(19) BUNDESREPUBLIK DEUTSCHLAND

DEUTSCHES PATENTAMT

(12) **Offenlegungsschrift**

(10) **DE 196 00 081 A 1**

(51) Int. Cl.[6]:
G 06 F 12/16
G 06 F 17/60
G 06 K 19/07

(21) Aktenzeichen: 196 00 081.5
(22) Anmeldetag: 3. 1. 96
(43) Offenlegungstag: 17. 7. 97

(71) Anmelder:
International Business Machines Corp., Armonk, N.Y., US

(74) Vertreter:
Schäfer, W., Dipl.-Ing., Pat.-Anw., 70188 Stuttgart

(72) Erfinder:
Bublitz, Herrmann R., Dipl.-Ing., 71034 Böblingen, DE; Hänel, Walter, Dipl.-Phys., 71088 Holzgerlingen, DE; Rindtorff, Klaus, Dipl.-Inform., 71093 Weil im Schönbuch, DE

(56) Entgegenhaltungen:
US 49 22 456

Prüfungsantrag gem. § 44 PatG ist gestellt

(54) Erhöhung der Datenintegrität bei Datenträgerkarten

(57) Vorgestellt wird ein Verfahren und ein System zur Erhöhung der Datenintegrität auf Chipkarten. Erfindungsgemäß wird eine Sequenz von Schreibvorgängen auf einer Chipkarte als Einheit definiert, für die die Integrität der zu schreibenden Daten sichergestellt wird. Bei dem Schreibvorgang kann es sich um das Schreiben von Daten in eine oder mehrere Dateien handeln. Vorzugsweise werden die Daten einer Integritätseinheit, d. h. eine Einheit von Daten, für die die Integrität sichergestellt werden soll, in einen Schattenspeicher geschrieben.
Das erfindungsgemäße Verfahren verringert die Wahrscheinlichkeit erheblich, daß durch einen Fehler beim Schreiben von Daten auf die Chipkarte die Integrität der Daten gestört wird. Daten können über die Grenzen von Schreib-Lesegeräten wiederhergestellt oder komplettiert werden. Das erfindungsgemäße Verfahren kann als Basis dazu dienen, Datenbestände außerhalb der Chipkarte konsistent mit den Daten auf der Chipkarte zu halten.

DE 196 00 081 A 1

DE 196 00 081 A 1

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

BUNDESDRUCKEREI 05. 97 702 029/35

BNSDOCID: <DE_____19600081A1_ _>

Beschreibung

Gebiet der Erfindung

Die Erfindung betrifft die Datenintegrität bei Anwendungen mit Datenträgerkarten.

Stand der Technik

Gesicherte Datenträgerkarten oder Chipkarten (Smart Cards) finden heutzutage Anwendung in einer Reihe unterschiedlicher Funktionen und werden mit steigenden Stückzahlen verbreitet. Zum Teil werden auf den Chipkarten Daten von beträchtlichem Wert dargestellt, wie Zahlungsmittel, Zahlungsmittelrahmen oder persönliche (z. B. medizinische) Daten. Das Konzept der gesicherten Chipkarte als verteilter persönlicher Datenspeicher und zum Teil auch datenschutzrechtliche Gründe verhindern und erschweren eine Datensicherung. Aus diesen Gründen muß an die Integrität der Daten auf der Chipkarte höchste Ansprüche gestellt werden. Bei den zu erwartenden hohen Stückzahlen in der Verbreitung von Chipkarten führen selbst niedrige Fehlerraten zu beträchtlichen Mengen an Fehlern.

Unter der Integrität von Daten versteht man die Fehlerfreiheit der Daten in dem Sinne, daß sie die ihnen zugrunde gelegten Informationen als richtiges Abbild der Realität beinhalten und damit für eine sinnvolle Datenverarbeitung verwendet werden können. Entsprechend bedeutet eine Konsistenz der Daten die Übereinstimmung der Daten mit der Realität, die sie repräsentierten. Dazu ist sowohl eine formale (physische Beschaffenheit) als auch eine inhaltliche (semantische, funktionelle) Voraussetzung zu erfüllen.

Die Integrität der Daten auf der Chipkarte ist gefährdet insbesondere durch physikalische Einflüsse sowohl auf die Chipkarte als solches als auch während eines Schreibvorgangs.

Die meisten verwendeten Chipkarten implementieren den ISO-Standard 7816-X oder deren analoge nationalen Standards. Daten auf der Chipkarte sind demgemäß in Dateien organisiert. Teile einer Datei können als Datensatz oder als beliebige Zeichenfolge mit Schreibkommandos geschrieben werden. Fehler, die während des Schreibvorgangs in der Chipkarte auftreten, werden an das Schreib-Lesegerät gemeldet. Wird der Schreibvorgang aus mechanischen, elektrischen oder elektronischen Gründen nicht beendet und das Schreib-Lesegerät und/oder die Chipkarte kann keine Fehlerbehebung einleiten, sind die Daten inkonsistent.

Insbesondere werden Chipkarten immer häufiger zusammen mit nicht motorgetriebenen Chipkartenlesern eingesetzt. Speziell bei diesen Billiglesern, bei denen das Ein- und Ausführen der Chipkarte manuell geschieht, kann der Benutzer der Chipkarte den Vorgang durch vorzeitige Entnahme der Chipkarte unterbrechen. Auch äußere Einflüsse wie Erschütterungen oder Stromausfall können zu solchen Unterbrechungen der laufenden Anwendung fuhren. Dies kann zur Folge haben, daß die unterbrochene Anwendung inkonsistente Datenbestände auf der Chipkarte und im Chipkartenleser zurückläßt, die die Chipkarte für die weitere Anwendung unbrauchbar macht.

Auf einer Chipkarte seien beispielsweise 2 Funktionen untergebracht: eine elektronische Börse und eine Berechtigung zum Besuch eines Fitneßstudios. Jeden Monat wird die Berechtigung durch das Ausbuchen der Gebühr aus der Börse verlängert. Dieser Vorgang kann aus den folgenden Schritten bestehen:

- Abbuchen des Betrages aus der Börse
- Bestätigung der Abbuchung durch die Chipkarte und
- Verlängerung der Berechtigung.

Wird die Chipkarte nun zwischen den ersten beiden Schritten entnommen, erhält man einen inkonsistenten Zustand: Der Betrag ist abgebucht, aber die Berechtigung nicht verlängert. Da der Chipkartenleser auch die Bestätigung der Abbuchung nicht erhalten hat, zeigen seine Daten an, daß er weder abgebucht noch die Berechtigung verlängert hat. Damit ist die Chipkarte unbrauchbar und kann auch nicht mehr in einen konsistenten Zustand gebracht werden.

Die im Stand der Technik bekannten Verfahren zum Schreiben von Daten auf Chipkarten liefern keine Möglichkeit die Integrität der Daten zu gewährleisten, insbesondere wenn der Schreibvorgang während des Schreibens abgebrochen wird.

Zusammenfassung der Erfindung

Es ist Aufgabe der Erfindung ein Verfahren zur Erhöhung der Datenintegrität auf Chipkarte zu erreichen. Die Aufgabe der Erfindung wird durch die unabhängigen Ansprüche gelöst.

Erfindungsgemäß wird eine Sequenz von Schreibvorgängen auf einer Chipkarte als Einheit definiert, für die die Integrität der zu schreibenden Daten sichergestellt wird. Bei dem Schreibvorgang kann es sich um das Schreiben von Daten in eine oder mehrere Dateien handeln.

Eine Integritätseinheit, d. h. eine Einheit von Daten, für die die Integrität sichergestellt werden soll, wird vorzugsweise durch eine Kennung identifiziert. Der Start einer Integritätseinheit wird vorzugsweise mit einem definierten Startkommando an die Chipkarte implementiert, oder alternativ dazu durch eine Kennung in jedem Schreibkommando festgelegt. Das Ende der Integritätseinheit wird vorzugsweise mit einer Sequenz von Kommandos angezeigt, die bei einem erfolgreich durchgeführten Übertragungsvorgang zu einem festen Schreiben der Daten in einem Speicher der Chipkarte führen oder bei einem nicht erfolgreichen Übertragungsvorgang den Schreibvorgang abbrechen.

Vorzugsweise werden die Daten einer Integritätseinheit in einen Schattenspeicher geschrieben. Schattenspeicher können beispielsweise implementiert werden durch komplette Kopien der Dateien, in die die Daten geschrieben werden sollen oder durch Kopieren des Datenteils einer Datei, der beschrieben werden soll, wobei der Dateikopf einen Verweis auf die Originaldaten erhält. Weiterhin können Schattenspeicher durch Kopien von Dateisätzen, in die die Daten gespeichert werden sollen, in satzorientierten Dateien, wobei der Dateikopf Verweise auf aufeinanderfolgende Sätze enthält, implementiert werden.

Vorzugsweise wird das Ende einer Integritätseinheit mit einer Sequenz von zwei Kommandos wie z. B. "Schreiben vorbereiten" und "Schreiben", bzw. durch ein Kommando, das ein Abbrechen des Schreibvorgangs der Integritätseinheit bewirkt, wie z. B. "Schreiben abbrechen", angezeigt. Mit dem Kommando "Schreiben vorbereiten" werden die Verweise auf die neuen Daten so vorbereitet, daß mit einem Minimum an benötigten Schreibaufwand die Gültigkeit der neuen Daten festgelegt werden kann. Dies kann z. B. durch ein Anlegen eines Datenverzeichnisses erfolgen, welches die Verweise auf die jeweils gültigen Daten enthält. Mit dem Kommando "Schreiben" wird die Gültigkeit der neuen Daten festgelegt, eine Rückmeldung an das Schreib-Lesegerät gegeben und die neuen Daten als gültige Daten festlegt. Der Speicher für die bisher gültigen Daten kann freigegeben und beispielsweise als ein weiterer Schattenspeicher verwendet werden. Es ist zu verstehen, daß die Zuordnung eines Speicherbereiches zum Schattenspeicher nicht permanent sein muß, sondern daß die Zuordnung nach Bedarf, als ein dynamischer Prozeß, definiert werden kann.

Durch das erfindungsgemäße Verfahren wird die Wahrscheinlichkeit, daß Fehler die Integrität der Daten stören können wesentlich reduziert.

Mit dem entsprechenden Kommando, das ein Abbrechen des Schreibvorgangs der Integritätseinheit bewirkt, wie z. B. "Schreiben abbrechen", wird der Schattenspeicher zu einer erneuten Benutzung freigeben. Die originalen Daten bleiben nach wie vor zur Verfügung.

Ein weiteres Kommando, wie beispielsweise "Integritätsstatus", dient vorzugsweise dazu, den Status der Integritätseinheit zu erfragen, d. h. beispielsweise ob und welche Integritätseinheit noch nicht beendet ist. Aufgrund dieser Information kann die Integritätseinheit, z. B. nach einem Fehler in dem Schreib-Lesegerät abgebrochen oder in einem anderen Schreib-Lesegerät geordnet beendet werden.

Das erfindungsgemäße Verfahren verringert die Wahrscheinlichkeit erheblich, daß durch einen Fehler beim Schreiben von Daten auf die Chipkarte die Integrität der Daten gestört wird. Daten können über die Grenzen von Schreib-Lesegeräten wiederhergestellt oder komplettiert werden. Das erfindungsgemäße Verfahren kann als Basis dazu dienen, Datenbestände außerhalb der Chipkarte konsistent mit den Daten auf der Chipkarte zu halten.

Als Integritätseinheiten können vorzugsweise sogenannte Transaktionen, wie sie aus dem Bereich von Datenbankensystemen bekannt sind, definiert werden.

Weitere, vorteilhafte Ausführungen der Erfindung finden sich in den Unteransprüchen.

Detaillierte Beschreibung der Erfindung

Die Datenbestände der einzelnen Chipkartenfunktionen und die Daten des Lesegerätes, bzw. des treibenden Rechners, bilden — abstrakt gesehen — eine verteilte Datenbank. Eine Methode um verteilte Datenbanken konsistent zu halten, sind Transaktionen. Als eine Integrationseinheit, d. h. als eine Einheit von Daten, für die die Integrität sichergestellt werden soll, wird eine Transaktion festgelegt. Transaktionen sind insbesondere gekennzeichnet durch 4 Eigenschaften. Transaktionen sind:

— unteilbar, d. h. sie werden entweder vollständig oder gar nicht ausgeführt, was durch ein "two phase commit" (Zwei-Phasen-Festschreiben) Verfahren erreicht werden kann;
— konsistenzerhaltend, d. h. die Daten eines transaktionsgeschützten Datensatzes sind in sich stimmig, was unter Zuhilfenahme des "two phase commit" Verfahrens erreicht wird;
— isoliert, d. h. parallel ablaufende Transaktionen beeinflussen sich gegenseitig nicht, was durch ein Blockieren der von einer Applikation benötigten Daten für andere Applikationen bewerkstelligt werden kann;
— dauerhaft, d. h. die Dauerhaftigkeit der Änderungen an einem Datenbestand ist bei der Chipkarte direkt gekoppelt mit der Dauerhaftigkeit der in einem beschreibbaren, nicht flüchtigen Speicher (z. B. einem EEPROM) der Chipkarte gespeicherten Daten, wodurch die Dauerhaftigkeit für die Daten auf der Chipkarte i.a. ausreichend ist.

Die Problematik der isoliert bzw. parallel ablaufenden Transaktionen kann auf der Chipkarte auch dadurch umgangen werden, daß immer nur eine Transaktion gleichzeitig zugelassen wird.

Das erfindungsgemäße "two phase commit" Verfahren für Chipkartendaten garantiert, daß entweder alle Aktionen auf der Chipkarte innerhalb einer Transaktion oder keine ausgeführt werden. Um dieses "two phase commit" Verfahren zu implementieren, wird definiert:

— ein Anfang der Transaktion, entweder implizit durch
  — ein Anlegen einer Versorgungsspannung ("Power on") an die Chipkarte,
  — einen Schreibzugriff auf transaktionsgeschützte Daten, oder explizit
  — durch einen vorgegebenen Transaktionsbefehl wie beispielsweise: "Transaktion beginnen" oder "start transaction";
— eine Reihe von Transaktionssteuerungsbefehlen, wie z. B.:
  — "Schreiben vorbereiten" oder "prepare to commit" (Vorbereiten auf das Ende der Transaktion)
  — "Schreiben" oder "commit" (Ende-der Transaktion)

— "Schreiben abbrechen" oder "unroll" (Abbruch der Transaktion).

Die Chipkarte muß weiterhin in der Lage sein, Änderungen an einem Datenbestand seit dem Zeitpunkt des Beginns der Transaktion wieder rückgängig zu machen. Dies kann dadurch erreicht werden, indem ein Datenbestand, der verändert werden soll, in einen Schattenspeicher dupliziert wird und die Änderungen nur an einer Hälfte der Daten, also entweder der Kopie oder dem Original, ausgeführt werden. Vorzugsweise werden Änderungen nur an der im Schattenspeicher abgelegten Kopie durchgeführt. Wurde die Transaktion erfolgreich durchgeführt, wird die veränderte Hälfte der Daten zum gültigen Teil erklärt. Wurde die Transaktion nicht erfolgreich durchgeführt, also z. B. abgebrochen, bleibt die unveränderte Hälfte gültig.

Zur permanenten Speicherung veränderbarer Daten werden in Chipkarten vorzugsweise EEPROM Speicher (Electrical Erasable Programmable Read Only Memory) als beschreibbare, nicht flüchtige Speicher eingesetzt, die sich insbesondere durch folgende Eigenschaften auszeichnen:

— ein Löschen von Daten, d. h. ein Zurücksetzen der einzelnen Speicherzellen des EEPROM in einen inaktiven Zustand, ist nur in größeren Einheiten, sogenannten "Pages" (z. B. 32 Bytes) möglich (die Größe dieser Einheit wird meist durch den Hersteller vorgegeben);
— ein Setzen von Daten ist bitweise möglich, d. h. ein Aktivieren der einzelnen Speicherzellen ist unabhängig von den anderen Speicherzellen innerhalb der gewählten "Page" durchführbar;
— die Anzahl der Lösch-Schreibzyklen ist jedoch durch die Lebensdauer des EEPROM limitiert (Größenordnung 10**4 Lösch-Schreibzyklen), ein überproportional häufiges Schreiben auf eine Speicheradresse muß daher vermieden werden.

In anderen Worten ist ein Löschen nur kollektiv für eine gesamte "Page" möglich, während Daten innerhalb der "Page" individuell gesetzt werden können. Ein Verändern von Daten erfolgt im allgemeinen durch ein Löschen der "Pages" auf denen sich die Daten zusammen mit anderen Datensätzen befinden und ein anschließendes Schreiben der veränderten Daten und der Datensätze die nicht verändert werden sollten. Wird dieser Vorgang z. B. durch Stromausfall unterbrochen, sind alle Datensätze auf der "Page" ungültig. Dies zeigt, daß Datensätze, die auf einer gemeinsamen "Page" liegen, sich gegenseitig beeinflussen können.

Die Eigenschaft des EEPROM, daß ein Löschen von Daten nur in größeren Einheiten, den sogenannten "Pages" möglich ist, muß bei einem Duplizieren von Daten im EEPROM Speicher berücksichtigt werden. Es ist darauf zu achten, daß die beiden Hälften (Original und Kopie im Schattenspeicher) des Datenbestandes nicht auf derselben "Page" liegen, damit eine gegenseitige Beeinflussung ausgeschlossen werden kann.

Bei größeren Datenbeständen können die Daten in Transaktionseinheiten unterteilt werden. Dies hat den Vorteil, daß nur die Transaktionseinheiten kopiert werden müssen, die auch verändert werden. Ist gewährleistet, daß von einem größerem Datenbestand, der in Einheiten aufgeteilt ist, pro Transaktion nur eine Einheit verändert wird, genügt es für diese Einheit nur einen Schattenspeicher anzulegen. Es muß daher weniger Speicherplatz für den Schattenspeicher zur Verfügung gestellt werden.

Damit Kopie und Original auseinander gehalten werden können, stattet man sie vorzugsweise mit je einem Zähler aus. Beim Erstellen der Kopie wird der Zähler beispielsweise um 1 erhöht. Damit wird die Aktualität der Daten durch den Zählerstand repräsentiert (höherer Zählerstand = neuere Daten). Die Verwendung der Zähler hat den Vorteil, daß keine Daten des Originals verändert werden müssen.

Da es nur zwei Versionen des Datensatzes (Original und Kopie) für die durchzuführende Transaktion gibt und die Zähler dieser Kopien sich nur um 1 unterscheiden, kann ein Modulo-Zähler verwendet werden. Dies vermeidet auch das Problem eines Zählerüberlaufes. Bei Verwendung z. B. eines Modulo-4 Zählers kann die ältere Kopie wie folgt bestimmt werden: wenn der Zählerstand der selektierten Version + 1, modulo 4 den Zählerstand der anderen Kopie ergibt, handelt es sich um die ältere Version des Datensatzes. Anhand der unten aufgeführten Tabellen 2 und 3 läßt sich beispielsweise bestimmen, welche Daten zu einem bestimmten Zeitpunkt gültig sind.

Zur Verwaltung der Zustände und Koordination der Transaktion wird vorzugsweise jeder Transaktionseinheit einer der folgende Zustände zugeordnet:

— "Data_Opened", d. h. diese Version der Transaktionseinheit wurde neu erstellt und ist für Veränderungen zugänglich;
— "Data_Prepared", d. h. an dieser Version der Transaktionseinheit sind alle Änderungen beendet, und es dürfen auch keine mehr vorgenommen werden;
— "Data_Committed", d. h. diese Version der Transaktionseinheit im Schattenspeicher wird zur neuen gültigen Version des Datensatzes erklärt;
— "Data_Unrolled", d. h. die Änderungen im Schattenspeicher werden für ungültig erklärt und der Schattenspeicher kann anderweitig verwendet werden.

Diese Zustände müssen durch Bitkombinationen im EEPROM Speicher repräsentiert werden. Bedingt durch das Löschen von "Pages" können unerwünschte Zwischenzustände entstehen, d. h. Bitkombinationen, die weder den Ausgangs- noch den Endzustand beschreiben. Um dies zu vermeiden, werden die Zwischenzustände im EEPROM vorzugsweise so repräsentiert, daß die Zustandsfolgen:

— Data_Opened, Data_Prepared, Data_Committed
— Data_Opened, Data_Prepared, Data_Unrolled
— Data_Opened, Data_Unrolled

nur ein Aktivieren von Datenelementen im EEPROM erfordern und somit kein Löschen erforderlich ist. Dies kann z. B. durch die folgende Repräsentation der Zustände, wie in Tabelle 1 dargestellt, erfolgen, wobei eine 1 für ein aktives Bit im EEPROM steht.

Tabelle 1

Beispiel einer Zustandstabelle einer Transaktion, wobei x "Zustand beliebig" bedeutet

Zustand	Repräsentation
'data_opened'	'000'
'data_prepared'	'001'
'data_committed'	'011'
'data_unrolled'	'1xx'

Beim Start einer neuen Transaktion muß von dem Zustand "data_unrolled" in den Zustand "data_opened" übergegangen werden. Hier ist nun ein Löschen der Repräsentation des Zustands im Speicher (vgl. Tabelle 1) nötig. Dabei können durch Unterbrechen des Löschvorgangs die unerwünschten Zwischenzustände "011" und "001" entstehen. Damit diese Zwischenzustände nicht zu einem ungewünschten Ergebnis führen, werden die gültigen Daten vor dem Löschvorgang in den Schattenspeicher kopiert. Somit entsprechen beide Datensätze (Originaldaten und Daten im Schattenspeicher) der gültigen Version und bei einem Auftreten von Zwischenzuständen kann eine beliebige der gültigen Versionen ausgewählt werden.

Zusätzlich kann noch ein globaler Merker verwendet werden, wie z. B. "Transaction_in_Process", der anzeigt, ob gerade eine Transaktion aktiv ist. Der globale Merker erlaubt somit ein Erkennen von Zwischenzuständen. Zwischenzustände können nur auftreten, wenn der globale Merker "transaction_in_process" anzeigt, daß keine Transaktion aktiv ist. Daher darf der globale Merker erst aktiv gesetzt werden, wenn die Kopie den Zustand "data_opened" hat.

Zähler und Zustandsanzeiger werden vorzugsweise auf von den Daten getrennten "Pages" untergebracht damit das Schreiben von Daten die Zustände nicht beeinflussen kann. Dies gilt auch für die Zähler und Zustandsanzeiger unterschiedlicher Versionen des Datensatzes.

Aus Zustand und Zählerstand beider Versionen des Datensatzes und dem globalen Merker "Transaktion in Bearbeitung" oder "transaction_in_process" kann zu jedem Zeitpunkt geklärt werden, welche Daten beim Start einer neuen Transaktion gültig sind. Die folgenden Tabellen 2 und 3 zeigen die möglichen Zuordnungen. Dabei-bedeutet "alt", daß diese Version des Datensatzes den kleineren Zählerstand hat.

Tabelle 2

Gültigkeitstabelle für den globalen Merker "transaction_in_process" aktiv

'alte' Daten	'neue' Daten	gültig
011	000	Gültig sind hier die 'alten' Daten. Die vorhergehende Transaktion wurde abgebrochen. Da der Zustand 'Data_Prepared' noch nicht erreicht wurde, können die neuen Daten gelöscht werden.
011	001	Die Gültigkeit der Daten muß von der Anwendung geklärt werden, da die vorhergehende Transaktion abgebrochen wurde.
011	011	Gültig sind hier die 'neuen' Daten. Die vorhergehende Transaktion war erfolgreich, aber die Transaktionsverarbeitung wurde vor einer Invalidierung der 'alten' Daten abgebrochen. Die 'alten' Daten können gelöscht werden
011	1xx	Gültig sind hier die 'alten' Daten. Die vorhergehende Transaktion wurde abgebrochen. Der Zustand 'Data_Enrolled' zeigt an, daß die 'neuen' Daten gelöscht werden können.

Tabelle 3

Gültigkeitstabelle für den globalen Merker "transaction_in_process" nicht aktiv

'alte' Daten	'neue' Daten	gültig
011	000	Gültig sind hier die 'alten' Daten. Die vorhergehende Transaktion wurde abgebrochen. Da der Zustand 'Data_Prepared' noch nicht erreicht wurde, können die neuen Daten gelöscht werden.
011	001	Gültig sind hier sowohl die 'alten' als auch die 'neuen' Daten. Der Zustand 001 kann außerhalb einer laufenden Transaktion nur als Zwischenzustand auftreten, in dem sowohl Original als auch Kopie gültig sind.
011	011	Gültig sind hier sowohl die 'alten' als auch die 'neuen' Daten. Der Zustand 011 kann außerhalb einer laufenden Transaktion nur als Zwischenzustand auftreten, in dem sowohl Original als auch Kopie gültig sind.
011	1xx	Gültig sind hier die 'alten' Daten. Die vorhergehende Transaktion wurde abgebrochen. Der Zustand 'Data_Enrolled' zeigt an, daß die 'neuen' Daten gelöscht werden können.

Da die Gültigkeit der Daten in besonderen Fällen nur mit Hilfe der Applikation geklärt werden kann, wird der Transaktion vorzugsweise ein eindeutiger Name zugeordnet, damit die Applikation das Protokoll des entsprechenden Transaktionsablaufes identifizieren kann. Dieser Transaktionsname kann z. B. aus einer Chip-Serien-

nummer und einer laufenden Transaktionsnummer bestehen.

Für eine Implementierung auf einer Chipkarte kann der Anfang einer Transaktion durch einen Transaktionsbefehl, wie z. B. den Befehl "start_transaction" bestimmt werden. Wird der Anfang einer Transaktion erkannt und ist noch keine Transaktion einer anderen Applikation in Bearbeitung, werden die gültigen Daten ermittelt und in den Schattenspeicher kopiert. Erst dann wird die Kopie in den Zustand "data_opened" gebracht. Ist dies erfolgreich durchgeführt worden, kann der globale Merker "transaction_in_process" gesetzt und der Applikation die laufende Transaktionsnummer mitgeteilt werden. Konnte die gültige Kopie nicht ermittelt werden, da die vorhergehende Transaktion Daten im Zustand "data_prepared" zurückgelassen hat, wird dies der Applikation mitgeteilt, damit diese klären kann, welche Daten gültig sind.

Ein geeignetes Chipkarten-Kommando zur Vorbereitung des Schreibvorgangs, z. B. "Schreiben vorbereiten" oder "Prepare to commit", bewirkt, daß aus dem Zustand "data_opened" in den Zustand "data_prepared" übergegangen wird. In allen anderen Zuständen ist dieser Befehl nicht erlaubt. Dies wird dies der Applikation mitgeteilt und es findet keine Zustandsänderung statt.

Ein weiteres geeignetes Chipkarten-Kommando zur Durchführung des Schreibvorgangs, z. B. "Schreiben" oder "commit", bewirkt, daß aus dem Zustand "data_repared" der Kopie in den Zustand "data_committed" übergegangen wird. In allen anderen Zuständen ist dieser Befehl ungültig und es findet keine Zustandsänderung statt.

Schließlich bewirkt ein anderes Chipkarten-Kommando zum Abbrechen des Schreibvorgangs, z. B. "Schreiben abbrechen" oder "unroll", daß aus dem Zustand "data_opened" oder "data_prepared" in den Zustand "data_unrolled" übergegangen wird. In allen anderen Zuständen ist das Kommando ungültig und es findet keine Zustandsänderung statt.

Tabelle 4 zeigt Beispiele für den Transaktionsablauf einer vollständigen Transaktion

Original			Kopie im Schatten- speicher		
Zustand	Zähler	Daten	Zustand	Zähler	Daten
Zustand vor Beginn der Transaktion					
011	1	ABCDEF	1xx	0	
Zustand nach dem Start der Transaktion					
011	1	ABCDEF	000	2	ABCDEF
Zustand nach einem Schreibzugriff					
011	1	ABCDEF	000	2	XYZDEF
Zustand nach 'prepare to commit'					
011	1	ABCDEF	001	2	XYZDEF
Zustand nach 'commit'					
011	1	ABCDEF	011	2	XYZDEF
111	1	ABCDEF	011	2	XYZDEF

Tabelle 5 zeigt Beispiele für den Transaktionsablauf einer abgebrochenen Transaktion

Original			Kopie im Schatten- speicher		
Zustand	Zähler	Daten	Zustand	Zähler	Daten
Zustand vor Beginn der Transaktion					
011	1	ABCDEF	1xx	0	
Zustand nach dem Start der Transaktion					
011	1	ABCDEF	000	2	ABCDEF
Zustand nach einem Schreibzugriff					
011	1	ABCDEF	000	2	XYZDEF
Unterbrechung der Transaktion Transaktion wird neu gestartet Zustand vor Neubeginn der Transaktion					
011	1	ABCDEF	000	2	XYZDEF
Zustand nach dem Start der Transaktion					
011	1	ABCDEF	000	2	ABCDEF
011	1	ABCDEF	101	2	XYZDEF
weiter mit normaler Transaktion					

Patentansprüche

1. Verfahren zur Sicherung der Integrität von Daten bei einer Kommunikation unter Beteiligung einer Datenträgerkarte, **dadurch gekennzeichnet,** daß eine Sequenz von Schreibvorgängen, für die die Integrität der zu schreibenden Daten sichergestellt werden soll, auf der Chipkarte als eine Integrationseinheit definiert wird.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die Integritätseinheit durch eine Kennung identifiziert wird.
3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß der Start der Integritätseinheit mit einem definierten Startkommando angezeigt wird.
4. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß der Start der Integritätseinheit durch eine Startkennung für ein jedes Schreibkommando zum Schreiben der Integritätseinheit festgelegt wird.
5. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß das Ende der Integritätseinheit mit einer Sequenz von Kommandos angezeigt wird, die bei einem erfolgreich durchgeführten Übertragungsvorgang zu einem festen Schreiben der Daten in einem Speicher der Chipkarte führen oder bei einem nicht erfolgreichen Übertragungsvorgang den Schreibvorgang abbrechen.
6. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß ein Schattenspeicher verwendet wird, in den Daten der Integritätseinheit geschrieben werden können.
7. Verfahren nach Anspruch 6, dadurch gekennzeichnet, daß der Schattenspeicher durch komplette Kopien der Dateien, in die die Daten geschrieben werden sollen oder durch Kopieren des Datenteils einer Datei, der beschrieben werden soll, wobei der Dateikopf einen Verweis auf die Originaldaten erhält, implementiert wird.
8. Verfahren nach Anspruch 6, dadurch gekennzeichnet, daß der Schattenspeicher durch Kopien von Dateisätzen, in die die Daten gespeichert werden sollen, in satzorientierten Dateien, implementiert wird, wobei der Dateikopf Verweise auf aufeinanderfolgende Sätze enthält.
9. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß
das Ende der Integritätseinheit im Falle einer erfolgreichen Übertragung mit einer Sequenz bestehend aus einem ersten und einem zweiten Kommando angezeigt wird, wobei das erste Kommando die Verweise auf

neue gültige Daten so vorbereitet, daß mit einem Minimum an benötigten Schreibaufwand die Gültigkeit der neuen Daten festgelegt werden kann, und das zweite Kommando die Gültigkeit der neuen Daten festlegt, eine Rückmeldung an das Schreib-Lesegerät gibt und die neuen Daten als gültige Daten festlegt; und

das Ende der Integritätseinheit im Falle einer nichterfolgreichen Übertragung durch ein Kommando, das ein Abbrechen des Schreibvorgangs der Integritätseinheit bewirkt, angezeigt wird.

10. Verfahren nach Anspruch 9, dadurch gekennzeichnet, daß das Festlegen der Gültigkeit der neuen Daten durch ein Anlegen eines Datenverzeichnisses erfolgt, welches Verweise auf die jeweils gültigen Daten enthält.

11. Verfahren nach Anspruch 9 oder 10, dadurch gekennzeichnet, daß der Speicher nach Durchführung der entsprechenden Kommandos für die bisher gültigen Daten freigegeben wird und als ein weiterer Schattenspeicher verwendet werden kann.

12. Verfahren nach einem der vorstehenden Ansprüche 6—11, dadurch gekennzeichnet, daß die Zuordnung des Speicherbereiches zum Schattenspeicher nach Bedarf, als ein dynamischer Prozeß, erfolgt.

13. Verfahren nach einem der vorstehenden Ansprüche 6—11, dadurch gekennzeichnet, daß Änderungen nur an den in dem Schattenspeicher geschriebenen Daten ausgeführt werden.

14. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß ein Status-Kommando durchgeführt wird, um den Status der Integritätseinheit zu erfragen, d. h. beispielsweise ob und welche Integritätseinheit noch nicht beendet ist.

15. Verfahren nach Anspruch 14, dadurch gekennzeichnet, daß aufgrund der aus dem Status-Kommando gewonnenen Information die Integritätseinheit bei einem Fehler abgebrochen wird.

16. Verfahren nach Anspruch 15, dadurch gekennzeichnet, daß die Integritätseinheit nach einem Fehler in dem Schreib-Lesegerät, in dem Schreibvorgang erfolgen sollte, abgebrochen und in einem anderen Schreib-Lesegerät geordnet beendet wird.

17. Verfahren entsprechend einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die Integritätseinheit eine Transaktion ist, die unteilbar, konsistenzerhaltend, isoliert und dauerhaft ist.

18. Verfahren entsprechend einem der vorstehenden Ansprüche 6—17, dadurch gekennzeichnet, daß ein Datenbestand, der verändert werden soll, in den Schattenspeicher dupliziert wird und Änderungen nur an einer Hälfte der Daten, also entweder der Kopie oder dem Original, ausgeführt werden.

19. Verfahren nach Anspruch 18, dadurch gekennzeichnet, daß die Änderungen nur an der in dem Schattenspeicher abgelegten Kopie durchgeführt werden.

20. Verfahren nach Anspruch 18 oder 19, dadurch gekennzeichnet, daß, wenn die Sequenz von Schreibvorgängen erfolgreich durchgeführt wurde, die veränderte Hälfte der Daten zum gültigen Teil erklärt wird, und wenn die Sequenz von Schreibvorgängen nicht erfolgreich durchgeführt wurde, die unveränderte Hälfte gültig bleibt.

21. Verfahren nach einem der Ansprüche 18 bis 20, dadurch gekennzeichnet, daß ein Modulo-Zähler zur Unterscheidung von Kopie und Original verwendet wird.

22. Verfahren nach einem der Ansprüche 18 bis 21, dadurch gekennzeichnet, daß eine Repräsentation einer Zustandsfolge benutzt wird, die es erlaubt nur durch Aktivieren von Speicherzellen die Zustandsfolgen zu durchlaufen.

23. Verwendung des Verfahrens entsprechend einem der vorstehenden Ansprüche, um Datenbestände außerhalb einer Chipkarte konsistent mit den Daten auf der Chipkarte zu halten.

24. System mit mindestens einer Datenträgerkarte, die geeignet ist Daten zu speichern, dadurch gekennzeichnet, daß bei einer Kommunikation unter Beteiligung der Datenträgerkarte zur Sicherung der Integrität von Daten eine Sequenz von Schreibvorgängen, für die die Integrität der zu schreibenden Daten sichergestellt werden soll, auf der Chipkarte als eine Integrationseinheit definiert wird.